Zenoss Foglight Agent

Custom Integrations

LightSpeed PM

Foglight® by Quest

Why Zenoss from LightSpeed PM?

Foglight requires custom services to integrate with Zenoss. Without our custom cartridge these steps must be taken:

- Custom Event Driven Rule
  - Command Invocation
  - Actions
- Extensive Groovy Scripting for more control over sent events and other important metrics
- Defining Zenoss Fields in Foglight

Zenoss Integration With LightSpeed PM’s Custom Cartridge

Advanced Rich UI for All Functionalities

Customizable REST invocation Support

- Create/Update tickets
- Ability to export data
- Flexible parameter definitions
- Auto Annotate Alarms with returned information

Customizable Command Line Invocation Support

Remote Administration of solution across servers

Decision Engine for total / precise control

- 20 levels of granularity
- Severity Level control

Zenoss – **Highlights**

Pattern Editor allows easy access to information related to the alarm

- No Groovy required
- Access to dozens of fields to easily provide values to parameters
- A dozen common operators available to easily avoid any coding

Use of Groovy scripts for power users

- Use of Groovy scripting with complete injected flow context
- Ability to reject/delay ticket generation

Persisted Queue so no events are lost

- Guaranteed delivery
- Keep history for rejected, failed and successful operations for total control
- Pause or delay control
- Consumer Thread settings/control to manage concurrency and volume

Zenoss Integration – **Alarming Features**

Auto Acknowledge Alarms

Alarm Flood Prevention

Alarm Transition Support

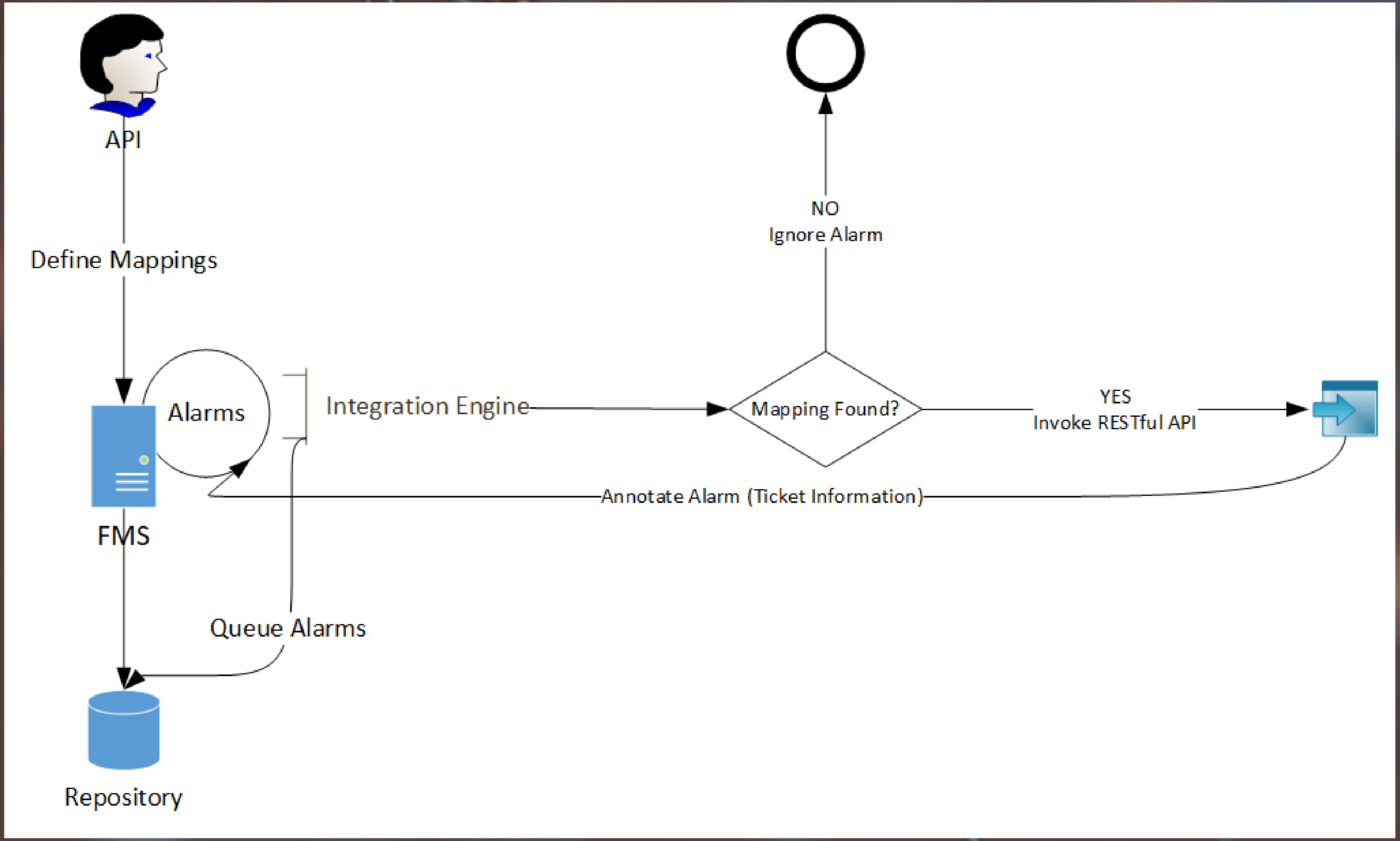
Can be Turned ON or OFF

Update tickets when alarm is transitioning from on**e** severity to another.
Can be either to higher severity only or any transition

Close ticket only when the last alarm in transition has been cleared

All transition information is accessible from the Pattern Editor

Zenoss Integration – RESTful API Flow

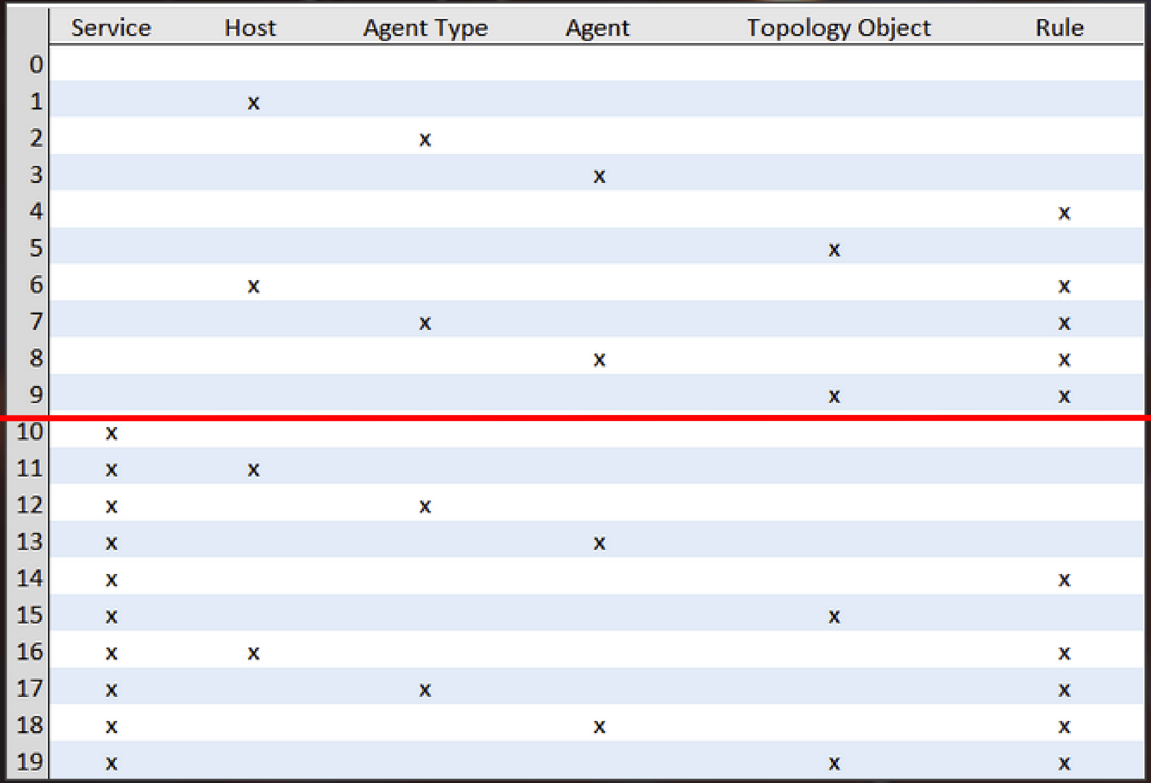


Zenoss Integration – **Decision Engine**

A rich UI within the solution presents the various mappings produced by the decision engine

Low Priority

No Service

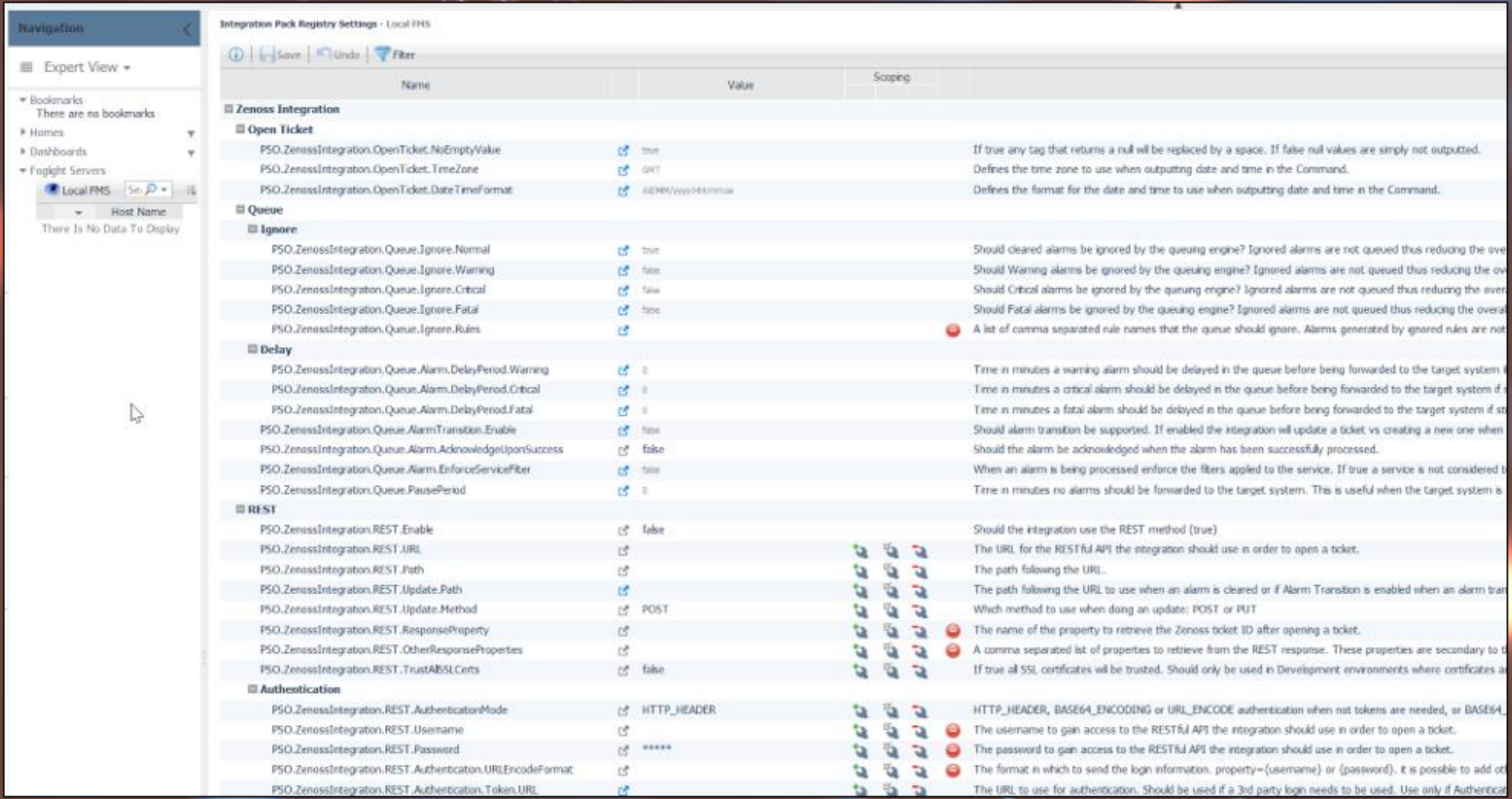


	Service	Host	Agent Type	Agent	Topology Object	Rule
0						
1		x				
2			x			
3				x		
4						x
5					x	
6		x				x
7			x			x
8				x		x
9					x	x
10	x					
11	x	x				
12	x		x			
13	x			x		
14	x					x
15	x				x	
16	x	x				x
17	x		x			x
18	x			x		x
19	x				x	x

High Priority

With Service

Zenoss Integration – **Changing Settings**

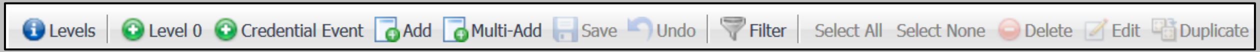


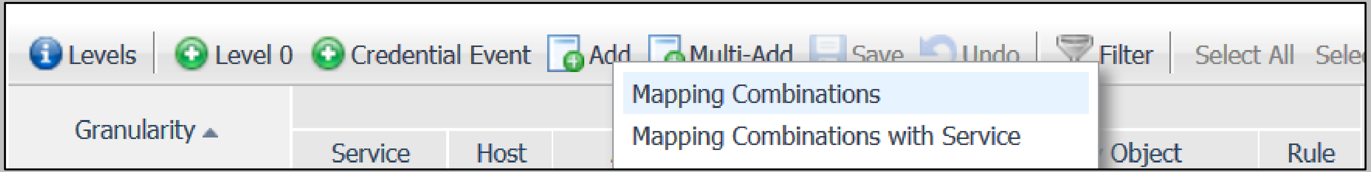
Integration Pack Registry Settings - Local FMS

Name	Value	Scoping	
Zenoss Integration			
Open Ticket			
PSO.ZenossIntegration.OpenTicket.NoEmptyValue	true		If true any tag that returns a null will be replaced by a space. If false null values are simply not outputted.
PSO.ZenossIntegration.OpenTicket.TimeZone	GMT		Defines the time zone to use when outputting date and time in the Command.
PSO.ZenossIntegration.OpenTicket.DateTimeFormat	dd/MM/yyyy HH:mm:ss		Defines the format for the date and time to use when outputting date and time in the Command.
Queue			
Ignore			
PSO.ZenossIntegraton.Queue.Ignore.Normal	true		Should cleared alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the ove
PSO.ZenossIntegraton.Queue.Ignore.Warning	false		Should Warning alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the ov
PSO.ZenossIntegraton.Queue.Ignore.Critical	false		Should Critical alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the over
PSO.ZenossIntegraton.Queue.Ignore.Fatal	false		Should Fatal alarms be ignored by the queuing engine? Ignored alarms are not queued thus reducing the overal
PSO.ZenossIntegraton.Queue.Ignore.Rules		●	A list of comma separated rule names that the queue should ignore. Alarms generated by ignored rules are not
Delay			
PSO.ZenossIntegraton.Queue.Alarm.DelayPeriod.Warning	0		Time in minutes a warning alarm should be delayed in the queue before being forwarded to the target system i
PSO.ZenossIntegraton.Queue.Alarm.DelayPeriod.Critical	0		Time in minutes a critical alarm should be delayed in the queue before being forwarded to the target system if s
PSO.ZenossIntegraton.Queue.Alarm.DelayPeriod.Fatal	0		Time in minutes a fatal alarm should be delayed in the queue before being forwarded to the target system if st
PSO.ZenossIntegration.Queue.AlarmTransition.Enable	false		Should alarm transition be supported. If enabled the integration will update a ticket vs creating a new one when
PSO.ZenossIntegraton.Queue.Alarm.AcknowledgeUponSuccess	false		Should the alarm be acknowledged when the alarm has been successfully processed.
PSO.ZenossIntegration.Queue.Alarm.EnforceServiceFilter	false		When an alarm is being processed enforce the filters applied to the service. If true a service is not considered t
PSO.ZenossIntegration.Queue.PausePeriod	0		Time in minutes no alarms should be forwarded to the target system. This is useful when the target system is
REST			
PSO.ZenossIntegration.REST.Enable	false		Should the integration use the REST method (true)
PSO.ZenossIntegration.REST.URL			The URL for the RESTful API the integration should use in order to open a ticket.
PSO.ZenossIntegration.REST.Path			The path following the URL.
PSO.ZenossIntegration.REST.Update.Path			The path following the URL to use when an alarm is cleared or if Alarm Transition is enabled when an alarm tran
PSO.ZenossIntegration.REST.Update.Method	POST		Which method to use when doing an update: POST or PUT
PSO.ZenossIntegration.REST.ResponseProperty		●	The name of the property to retrieve the Zenoss ticket ID after opening a ticket.
PSO.ZenossIntegration.REST.OtherResponseProperties		●	A comma separated list of properties to retrieve from the REST response. These properties are secondary to t
PSO.ZenossIntegration.REST.TrustAllSSLCerts	false		If true all SSL certificates will be trusted. Should only be used in Development environments where certificates a
Authentication			
PSO.ZenossIntegraton.REST.AuthenticationMode	HTTP_HEADER		HTTP_HEADER, BASE64_ENCODING or URL_ENCODE authentication when not tokens are needed, or BASE64_
PSO.ZenossIntegraton.REST.Username		●	The username to gain access to the RESTful API the integration should use in order to open a ticket.
PSO.ZenossIntegraton.REST.Password	*****	●	The password to gain access to the RESTful API the integration should use in order to open a ticket.
PSO.ZenossIntegraton.REST.Authentication.URLEncodeFormat		●	The format in which to send the login information. property={username} or {password}. it is possible to add ot
PSO.ZenossIntegraton.REST.Authentication.Token.URL			The URL to use for authentication. Should be used if a 3rd party login needs to be used. Use only if Authenticat

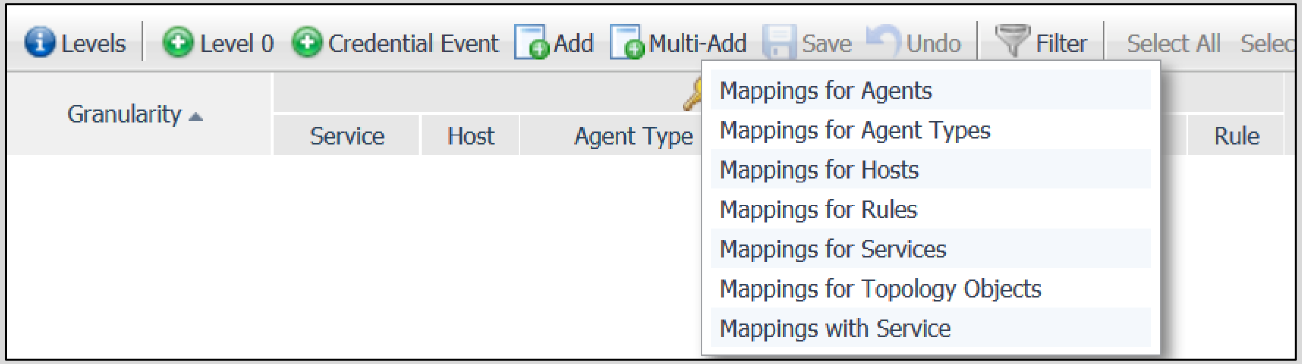
Zenoss Integration – Mappings Editor

Combinations

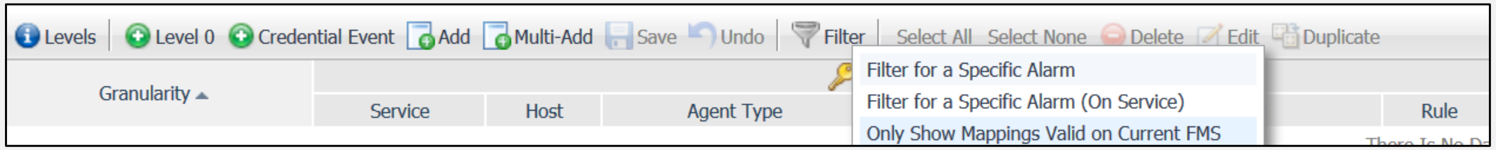


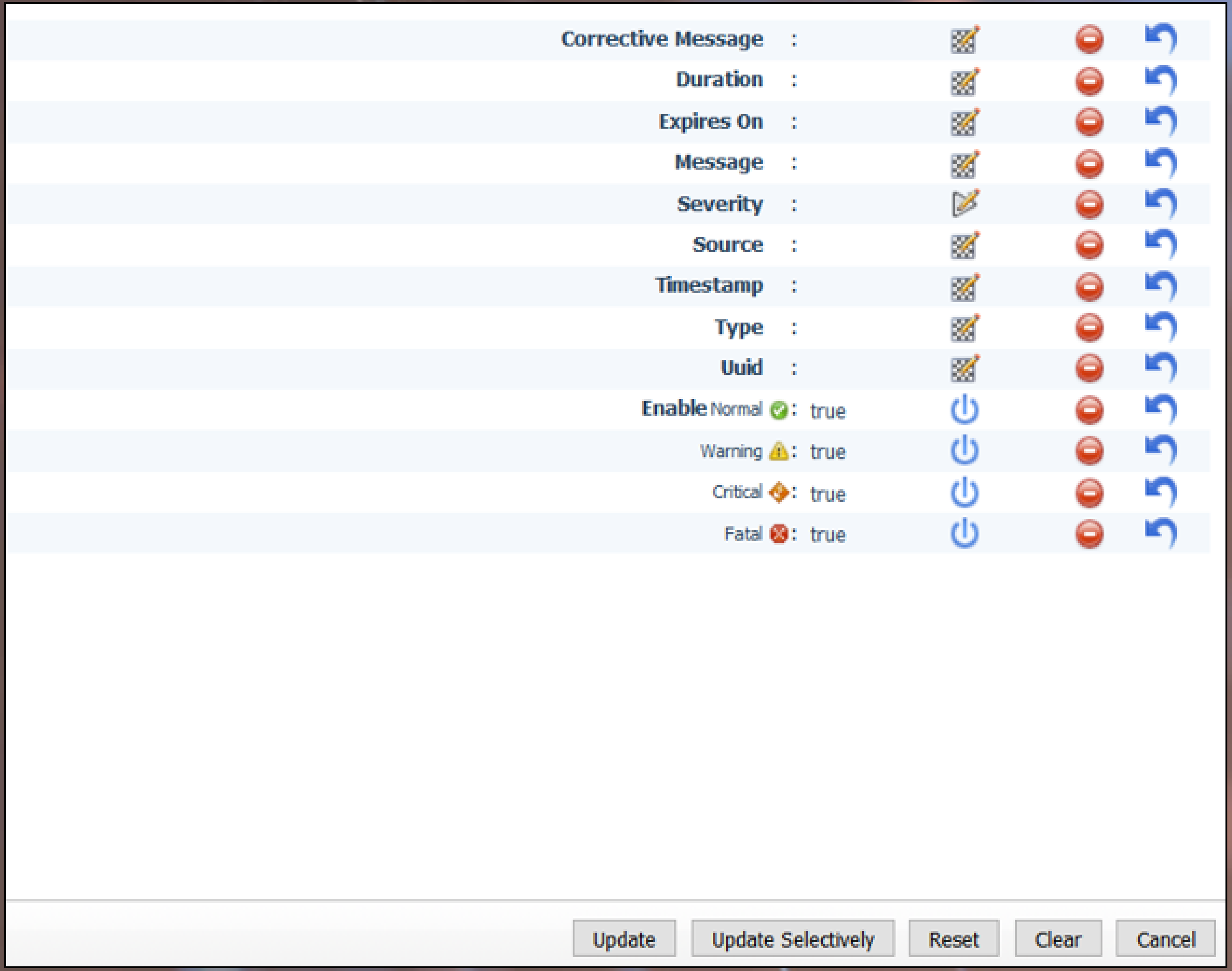


Multiple Mappings



Filtering





Corrective Message :

Duration :

Expires On :

Message :

Severity :

Source :

Timestamp :

Type :

Uuid :

Enable Normal ✓: true

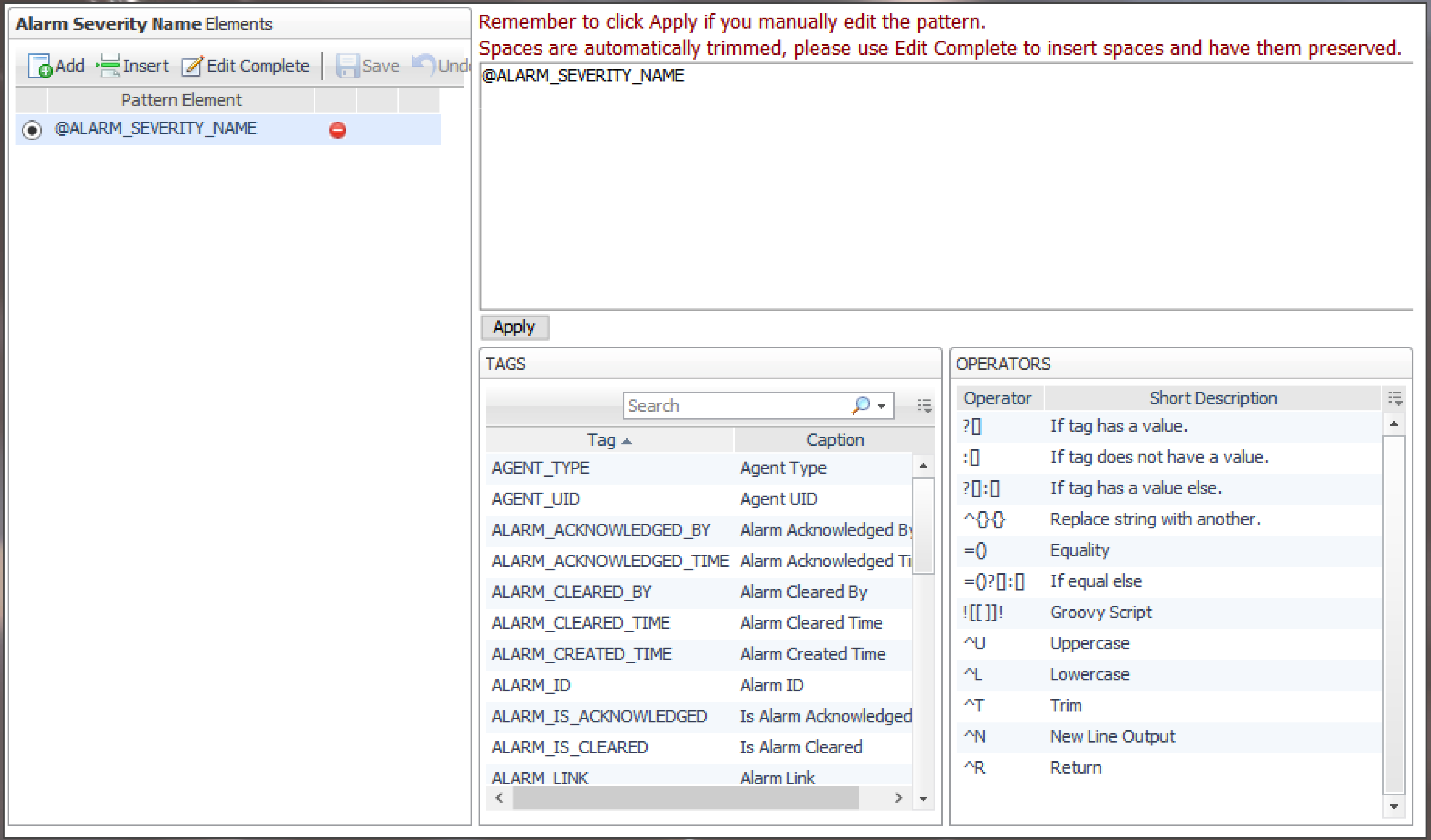
Warning ⚠: true

Critical ◆: true

Fatal ☒: true

Zenoss Integration – **Field Editor**

Zenoss Integration – **Pattern Editor**



Remember to click Apply if you manually edit the pattern.
Spaces are automatically trimmed, please use Edit Complete to insert spaces and have them preserved.

Pattern Element
● @ALARM_SEVERITY_NAME

```
@ALARM_SEVERITY_NAME
```

Tag ▲	Caption
AGENT_TYPE	Agent Type
AGENT_UID	Agent UID
ALARM_ACKNOWLEDGED_BY	Alarm Acknowledged By
ALARM_ACKNOWLEDGED_TIME	Alarm Acknowledged Ti
ALARM_CLEARED_BY	Alarm Cleared By
ALARM_CLEARED_TIME	Alarm Cleared Time
ALARM_CREATED_TIME	Alarm Created Time
ALARM_ID	Alarm ID
ALARM_IS_ACKNOWLEDGED	Is Alarm Acknowledged
ALARM_IS_CLEARED	Is Alarm Cleared
ALARM_LINK	Alarm Link

Operator	Short Description
?[]	If tag has a value.
:[]	If tag does not have a value.
?[]:[]	If tag has a value else.
^{}{}	Replace string with another.
=()	Equality
=()?[]:[]	If equal else
![[ ]]!	Groovy Script
^U	Uppercase
^L	Lowercase
^T	Trim
^N	New Line Output
^R	Return

Zenoss Integration – System Requirements

Minimum required FMS version 5.9.3

Supported Database Minimum Version

MS SQL	2008 v 10.0.1600 or later
Oracle	9i R2
MySQL	5.1.45
PostgreSQL	9.4.0

Performance Monitoring customized to your unique environment

Ten plus years of providing Professional Services to Quest customers revealed these enhancements to be most requested modifications.

Let us take Foglight’s out of the box capabilities and enhance for your unique environment.

Contact: Sales@LightSpeedPM.com